


@niallburkley | github.com/nburkley | niallburkley.com

# Elixir Deployment

@ Meltwater





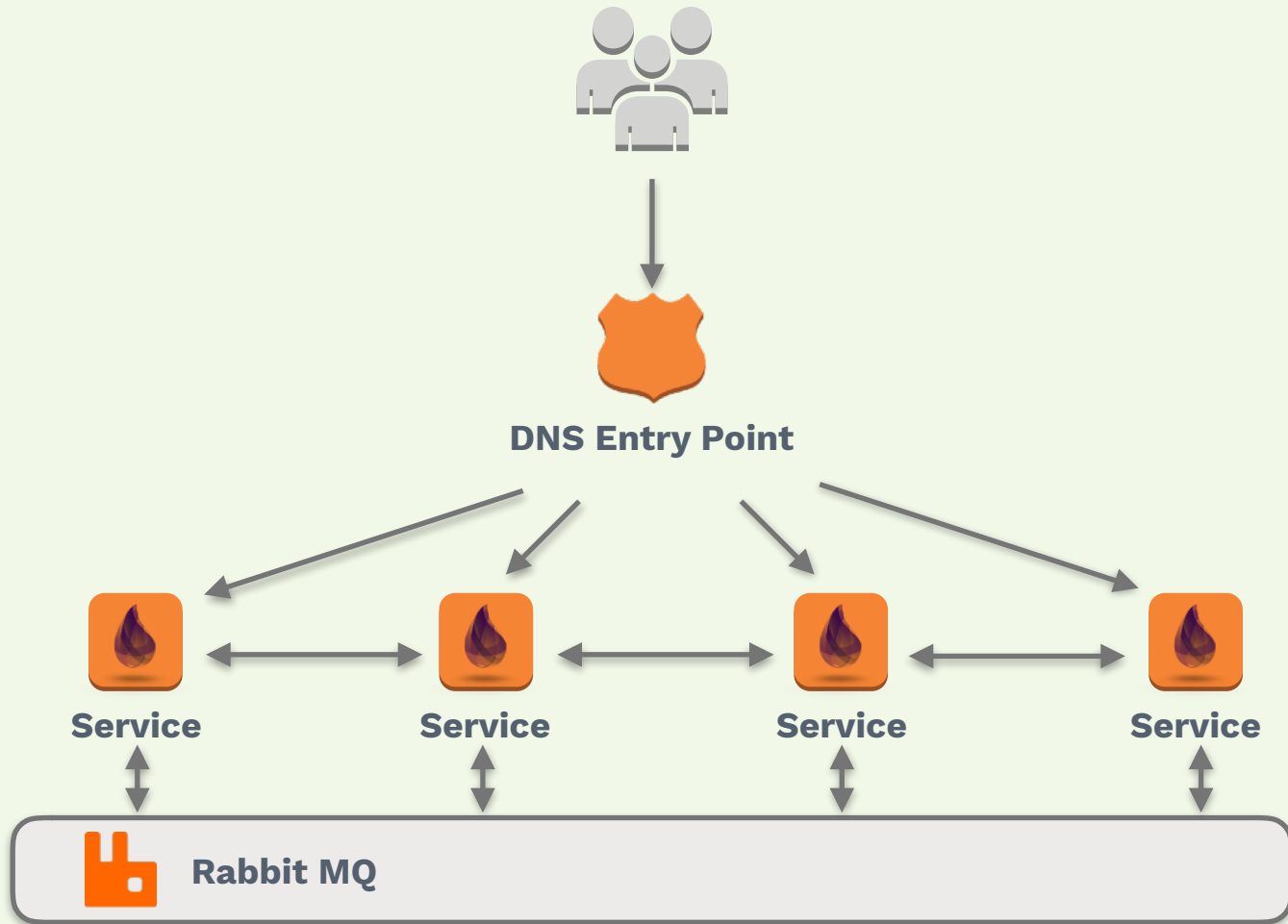


# How Elixir Deployment has Evolved

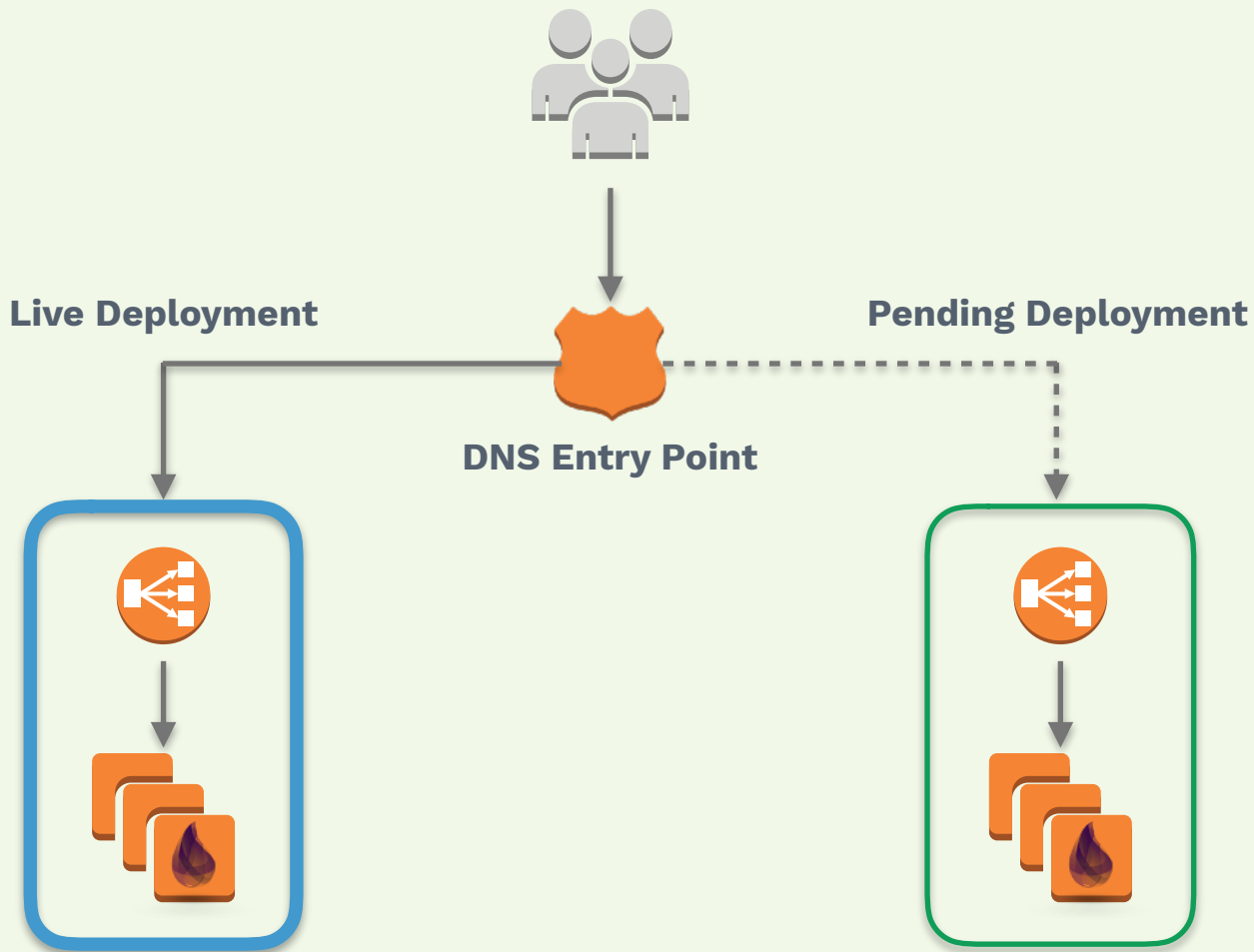
@ Meltwater

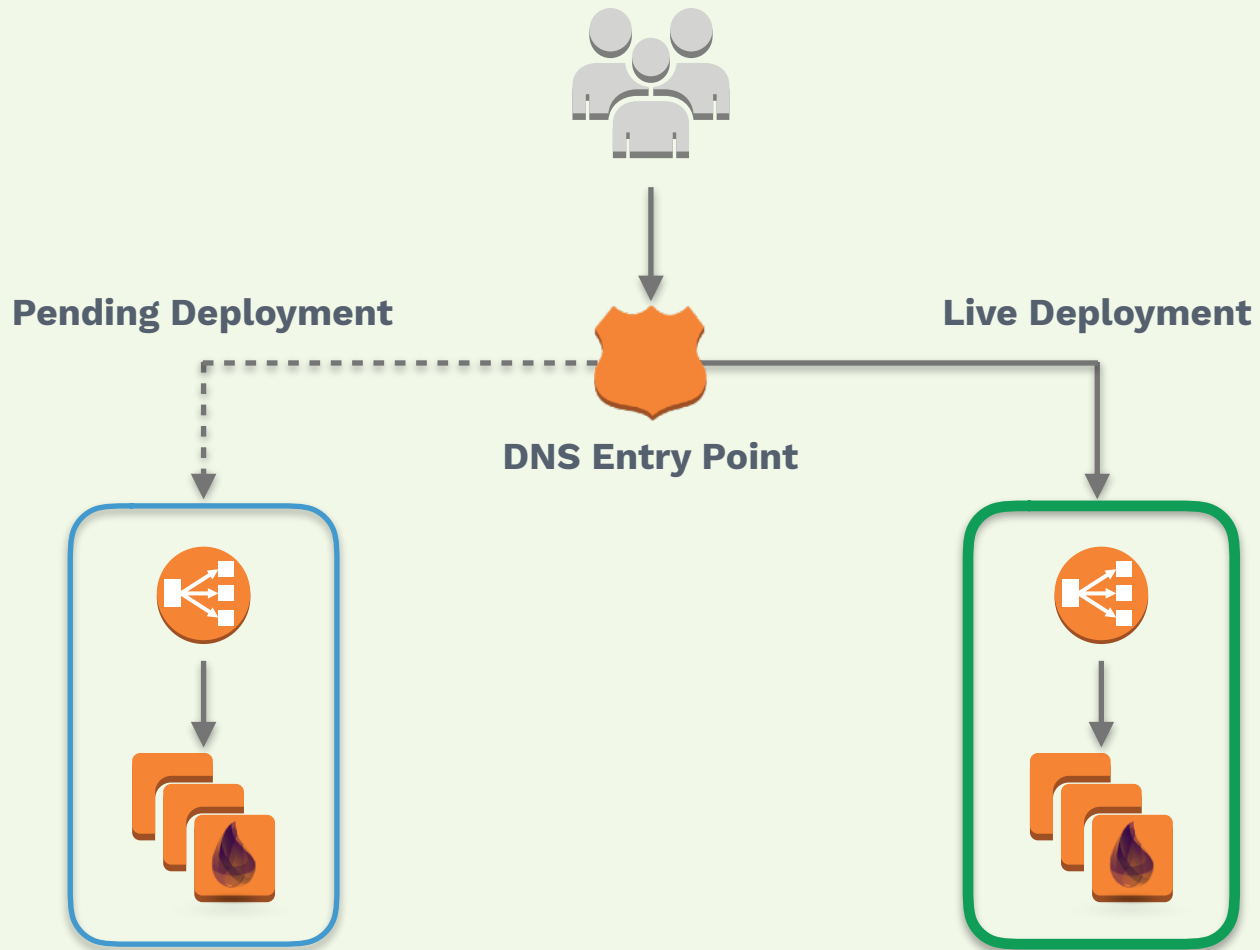
# Services

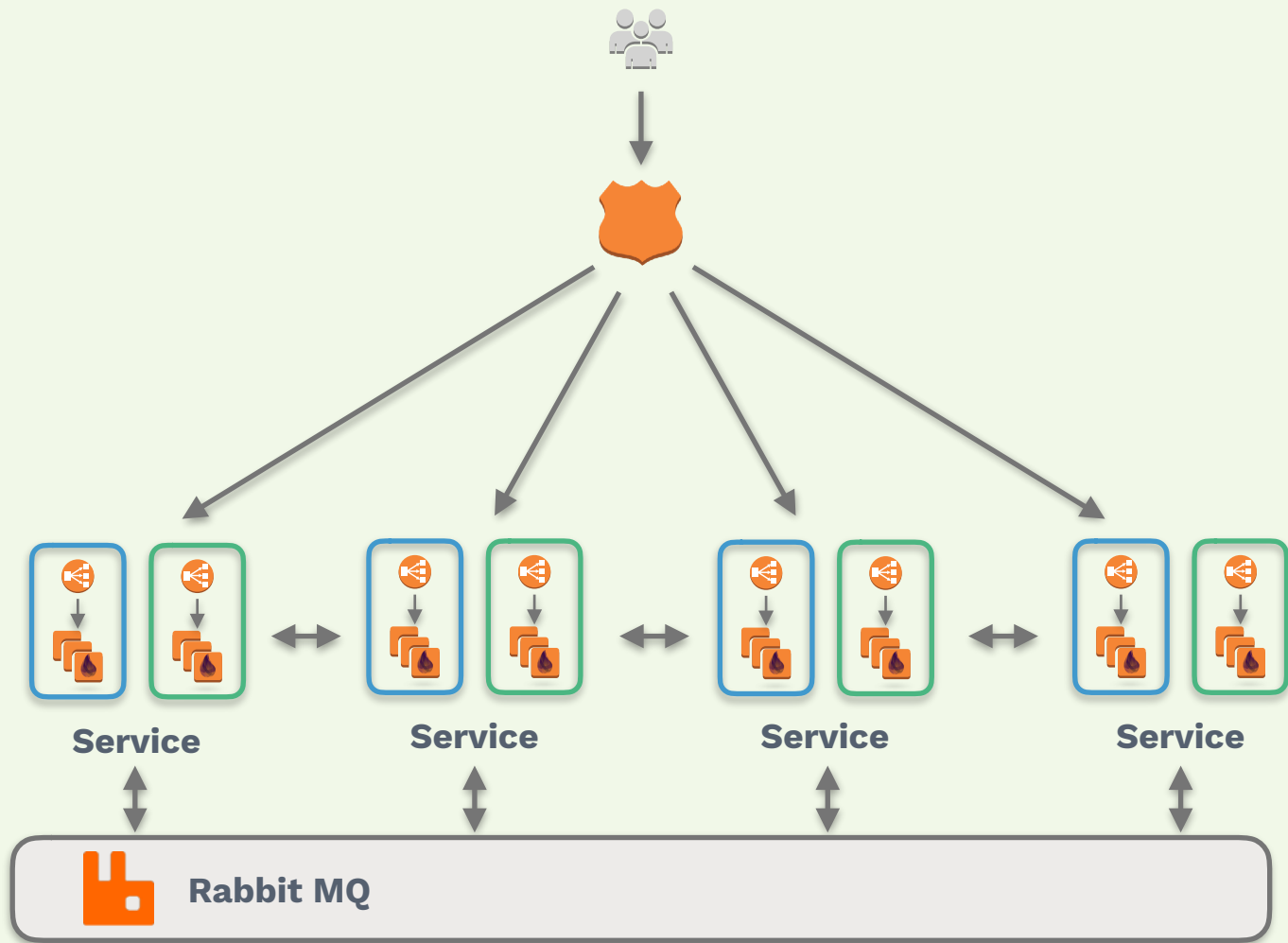
A photograph of several sailboats on a blue ocean under a grey sky. The word 'Services' is overlaid in large, bold, yellow-green letters. The sailboats have various colored sails, including red, white, blue, and yellow. One boat in the center has 'CAT' written on its hull.



# **Blue/Green Deployments**









# Travis CI + Elastic Beanstalk

# Travis CI -> Elastic Beanstalk: Config

```
# config.exe
# set runtime configs with environment variables
config :statix,
  host: System.get_env("STATSD_HOST"),
  port: (System.get_env("STATSD_PORT") |> String.to_integer

# set environment variables with Dockerrun.aws.json
"environment": [
  {
    "name": "STATSD_HOST",
    "value": "statsd.server.com"
  },
  {
    "name": "STATSD_PORT",
    "value": "8125"
  }
]
```

# Travis CI -> Elastic Beanstalk: Steps

# 1. install dependancies

```
mix deps.get
```

# 2. run tests

```
mix test
```

# 3. prepare docker image

```
docker build
```

# 4. deploy to elastic beanstalk

deploy:

- provider: elasticbeanstalk
- app: my-app
- env: my-app-green

# Travis CI -> Elastic Beanstalk: Dockerfile

```
FROM meltwater-private-repo/elixir-alpine:latest
```

```
ENV MIX_ENV prod
```

```
RUN mix local.hex --force
```

```
RUN mix local.rebar
```

```
RUN mix deps.get
```

```
RUN mix compile
```

```
RUN mix phx.swagger
```

```
ENTRYPOINT ["/launch.sh"]
```

# Travis CI -> Elastic Beanstalk: **launch.sh**

```
#!/bin/sh
```

```
set -e
```

```
# decrypt secrets
```

```
# https://github.com/meltwater/secretary
```

```
# source environment variables
```

```
echo "Starting application..."
```

```
MIX_ENV=prod elixir -sname my-app -S mix run --no-halt
```



## Travis CI -> Elastic Beanstalk - The Good 🍌

- ▶ Easy
- ▶ Blue/Green Deployments
- ▶ Scalable

## Travis CI -> Elastic Beanstalk - The Bad 🙄

- ▶ Slow to scale and deploy
- ▶ Slow to launch
- ▶ Poor use of resources
- ▶ Large Docker images

# **Drone CI + Kubernetes + Distillery**

# Drone



# Kubernetes





# Distillery 2.0

**Now with  
Config Providers!**

# Distillery & Drone CI -> Kubernetes: **config**

```
$> mix release init
```

- lib
- priv
- **rel**
- config
- plugins
- 🌀 config.exs
- test

# Distillery & Drone CI -> Kubernetes: **config**

```
# rel/config.exe
#
# Mix Config Provider
environment :prod do
  ...
  set(
    config_providers: [
      {Mix.Releases.Config.Providers.Elixir, ["${RELEASE_ROOT_DIR}/etc/config.exs"]}
    ]
  )
end
```

# Distillery & Drone CI -> Kubernetes: **config**

```
# rel/config/config.exe
# set runtime configs with environment variables
config :statix,
  host: System.get_env("STATSD_HOST"),
  port: (System.get_env("STATSD_PORT") |> String.to_integer
```

# Distillery & Drone CI -> Kubernetes: **config**

```
# set environment variables with k8_conf.yml
spec:
  containers:
  - name: my-app
    image: mw-private-docker-registry/mw-meltwater-app:latest
  env:
  - name: STATSD_HOST
    value: "statsd.server.com"
  - name: STATSD_PORT
    value: "8125"
```



# Distillery & Drone CI -> Kubernetes: **steps**

# 1. setup drone images

# 2. install dependencies

```
mix deps.get
```

# 3. run tests

```
mix test
```

# 4. prepare docker image

```
docker build
```

# 5. deploy to elastic beanstalk

```
deploy: image: meltwaterfoundation/drone-kubectl
```

```
  environment:
```

- NAME=my-app
- COLOR=green

```
  commands:
```

- kubectl apply -f k8s\_conf.yml --namespace \${NAMESPACE}

# Distillery & Drone CI -> Kubernetes: **Dockerfile**

```
# Multi-stage Docker Build
```

```
# Build in first stage
```

```
FROM elixir:1.7.3-alpine as builder
```

```
ENV MIX_ENV prod
```

```
RUN mix local.hex --force
```

```
RUN mix local.rebar
```

```
RUN mix deps.get
```

```
RUN mix compile
```

```
RUN mix release --env=${MIX_ENV} --no-tar
```

# Distillery & Drone CI -> Kubernetes: **Dockerfile**

```
# Multi-stage Docker Build
```

```
# Copy to second stage
```

```
FROM elixir:1.7.3-alpine as runner
```

```
COPY --from=builder /opt/mw-elixir-app/_build/prod/rel/my-app /opt/my-app
```

```
WORKDIR /opt/my-app
```

```
CMD ./bin/my-app foreground
```

**Distillery & Drone CI -> Kubernetes -> 👍**

- ▶ Build/Compile once
- ▶ Quick to deploy and scale
- ▶ Small docker images
- ▶ ‘The Elixir Way’

@niallburkley | [github.com/nburkley](https://github.com/nburkley) | [niallburkley.com](https://niallburkley.com)

# Thank you!



[underthehood.meltwater.com](https://underthehood.meltwater.com)